

IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A storage medium containing program instructions readable by a computer for detecting and resolving circular flow paths disposed within a flow diagram representing [[the]] logical operation of a corresponding application program, the flow diagram formed by interconnecting a plurality of symbolic representations of program objects, the program objects configured to execute associated functions in response to corresponding triggering events, the readable program instructions comprising program instructions for:

establishing a busy indicator at a given program object, the busy indicator signifying whether the given program object is currently executing its associated function during execution of the flow diagram;

in response to [[the]] an occurrence of a triggering event for the given program object ~~object's triggering event~~ during execution of the flow diagram, testing the respective busy indicator;

if the busy indicator signifies that the given program object is currently executing, blocking the given program object from re-executing in response to the triggering event;

if the busy indicator signifies that the given program object is not currently executing, permitting the given program object to execute in response to the triggering event.

2. (Original) The storage medium of claim 1 wherein the busy indicator is a counter and the program instructions for testing comprise program instructions for:

adjusting the counter; and

after the program instructions for adjusting, determining whether the counter exceeds a predetermined threshold,
wherein an exceedance of the predetermined threshold signifies that the given program object is currently executing.

3. (Original) The storage medium of claim 2 further comprising program instructions for initializing the counter to a null value, and wherein the program instructions for adjusting comprise program instructions for incrementing the counter.

4. (Original) The storage medium of claim 3 further comprising program instructions for, after the program instructions for determining whether the counter exceeds a predetermined threshold, decrementing the counter.

5. (Original) The storage medium of claim 4 wherein the program instructions for incrementing the counter increment the counter by 1, the program instructions for decrementing the counter decrement the counter by 1, and the predetermined threshold is 1.

6. (Original) The storage medium of claim 4 wherein the given program object includes one or more output properties having corresponding values that may be changed in response to execution of the given program object's associated function and, during execution, the given program object is configured to issue at least one ready event upon changing the values of its one or more output properties and one or more program objects may register for notification of the at least one ready event, further wherein the program instructions for decrementing the counter occur after all of the registered objects have been notified of the given object's at least one ready event.

7. (Previously Presented) A program object configured to execute an associated function in response to a triggering event, the program object used in developing an application program whose logical operation is represented by a corresponding flow diagram, the program object having program instructions for detecting and resolving circular flow paths disposed within the flow diagram, the program instructions comprising program instructions for:

establishing a busy indicator at the program object, the busy indicator signifying whether the program object is currently executing its associated function during execution of the flow diagram;

in response to an occurrence of the program object's triggering event during execution of the flow diagram, testing the busy indicator;

if the busy indicator signifies that the program object is currently executing, blocking the program object from re-executing in response to the triggering event;

if the busy indicator signifies that the program object is not currently executing, permitting the program object to execute in response to the triggering event.

8. (Original) The program object of claim 7 wherein the busy indicator is a counter and the program instructions for testing comprise program instructions for:

adjusting the counter; and

after the program instructions for adjusting, determining whether the counter exceeds a predetermined threshold,
wherein an exceedance of the predetermined threshold signifies that the program object is currently executing.

9. (Original) The program object of claim 8 further comprising program instructions for initializing the counter to a null value, and wherein the program instructions for adjusting comprise program instructions for incrementing the counter.

10. (Original) The program object of claim 9 further comprising program instructions for, after the program instructions for determining whether the counter exceeds a predetermined threshold, decrementing the counter.

11. (Original) The program object of claim 10 wherein the program instructions for incrementing the counter increment the counter by 1, the program instructions for decrementing the counter decrement the counter by 1, and the predetermined threshold is 1.

12. (Original) The program object of claim 10 wherein the program object includes one or more output properties having corresponding values that may be changed in response to execution of its associated function and, during execution, the program object is configured to issue at least one ready event upon changing the values of its one or more output properties and one or more other program objects within the application program may register for notification of the at least one ready event, further wherein the program instructions for decrementing the counter occur after all of the registered objects have been notified of the given object's at least one ready event.

13. (Currently Amended) A method for preventing circular flow paths in a ~~graphical programming system~~ flow diagram having a plurality of program objects, the method comprising:

- establishing a busy indicator at a given program object during execution of the flow diagram, the busy indicator signifying whether the given program object is currently executing an associated function;

- in response to an occurrence of a triggering event ~~for which causes~~ the given program object ~~object's associated function to execute~~ during execution of the flow diagram, testing the busy indicator;

- if the busy indicator signifies that the given program object is currently executing, blocking the given program object from re-executing in response to the triggering event;

- if the busy indicator signifies that the given program object is not currently executing, permitting the given program object to execute in response to the triggering event; and

- notifying other objects in the plurality of program objects when one or more output values associated with the given program object is changed during execution of the given program object's associated function.

14. (Currently Amended) An apparatus that prevents circular flow paths in a ~~graphical programming system~~ flow diagram having a plurality of program objects, the apparatus comprising:

means for establishing a busy indicator at a given program object during execution of the flow diagram, the busy indicator signifying whether the given program object is currently executing an associated function during execution of the flow diagram;

means for testing the busy indicator in response to an occurrence of a triggering event for which causes the given program object ~~object's associated function to execute~~;

means for blocking the given program object from re-executing in response to the triggering event if the busy indicator signifies that the given program object is currently executing; and

means for permitting the given program object to execute in response to the triggering event if the busy indicator signifies that the given program object is not currently executing. [[; and]]

~~means for notifying other objects in the plurality of program objects when one or more output values associated with the given program object is changed during execution of the given program object's associated function.~~

15. (Currently Amended) A method for preventing circular flow paths in a ~~graphical programming system~~ flow diagram having a plurality of program objects, the method comprising:

establishing a busy indicator at a given program object during execution of the flow diagram, the busy indicator signifying whether the given program object is currently executing an associated function during execution of the flow diagram;

in response to an occurrence of a triggering event for the given program object during execution of the flow diagram ~~which causes the given program object's associated function to execute~~, testing the busy indicator;

if the busy indicator signifies that the given program object is currently executing, blocking the given program object from re-executing in response to the triggering event; and

if the busy indicator signifies that the given program object is not currently executing, permitting the given program object to execute in response to the triggering event.

16. (Previously presented) The method of claim 15, further comprising:
 establishing the busy indicator as an adjustable counter; and
 determining whether the given program object is currently executing based on whether a value of the counter exceeds a predetermined threshold.
17. (Previously presented) The method of claim 16 further comprising:
 initializing the value of the counter to a null value.
18. (Previously presented) The method of claim 16 further comprising:
 incrementing the value of the counter when testing the busy indicator; and
 decrementing the value of the counter after determining whether the value of the counter exceeds a predetermined threshold.
19. (Previously presented) The method of claim 18 further comprising:
 setting the predetermined threshold equal to one;
 incrementing the value of the counter by one when the value of the counter is incremented;
 decrementing the value of the counter by one when the value of the counter is decremented.
20. (Previously presented) The method of claim 15, further comprising:
 issuing at least one ready event in response to one or more output values associated with the given object being changed during execution of the given program object's associated function; and
 notifying one or more program objects that the at least one ready event has been issued.
21. (Currently Amended) An apparatus for preventing circular flow paths in a ~~graphical programming system~~ flow diagram having a plurality of program objects, the apparatus comprising:

means for establishing a busy indicator at a given program object, the busy indicator signifying whether the given program object is currently executing an associated function during execution of the flow diagram;

means for testing the busy indicator in response to an occurrence of a triggering event for the given program object during execution of the flow diagram ~~which causes the given program object's associated function to execute~~;

means for blocking the given program object from re-executing in response to the triggering event if the busy indicator signifies that the given program object is currently executing; and

means for permitting the given program object to execute in response to the triggering event if the busy indicator signifies that the given program object is not currently executing.

22. (Currently Amended) A computer-readable media, comprising:

instructions for execution in a processor for the practice of a method for preventing circular flow paths in a ~~graphical programming system~~ flow diagram having a plurality of program objects, said method having the steps,

establishing a busy indicator at a given program object during execution of the flow diagram, the busy indicator signifying whether the given program object is currently executing an associated function during execution of the flow diagram;

in response to an occurrence of a triggering event for the given program object during execution of the flow diagram ~~which causes the given program object's associated function to execute~~, testing the busy indicator;

if the busy indicator signifies that the given program object is currently executing, blocking the given program object from re-executing in response to the triggering event; and

if the busy indicator signifies that the given program object is not currently executing, permitting the given program object to execute in response to the triggering event.

23. (Canceled)

24. (New) The storage medium of claim 1,

wherein establishing the busy indicator comprises establishing a counter for the given program object; and

wherein testing the busy indicator comprises testing a value of the counter.

25. (New) The program object of claim 7,

wherein establishing the busy indicator comprises establishing a counter for the program object; and

wherein testing the busy indicator comprises testing a value of the counter.

26. (New) The method of claim 15,

wherein establishing the busy indicator comprises establishing a counter for the program object; and

wherein testing the busy indicator comprises testing a value of the counter.